PREPRINT - LEARNING UNIONS OF ORTHONORMAL BASES WITH THRESHOLDED SINGULAR VALUE DECOMPOSITION

LESAGE Sylvain, GRIBONVAL Rémi, BIMBOT Frédéric, BENAROYA Laurent ⋆

IRISA (CNRS & INRIA), campus de Beaulieu, 35042 RENNES cedex, FRANCE
sylvain.lesage@irisa.fr, remi.gribonval@irisa.fr, frederic.bimbot@irisa.fr

ABSTRACT

We propose a new method to learn overcomplete dictionaries for sparse coding. The method is designed to learn dictionaries structured as unions of orthonormal bases. The interest of such a structure is manifold. Indeed, it seems that many signals or images can be modeled as the superimposition of several layers with sparse decompositions in as many bases. Moreover, in such dictionaries, the efficient Block Coordinate Relaxation (BCR) algorithm can be used to compute sparse decompositions. We show that it is possible to design an iterative learning algorithm that produces a dictionary with the required structure. Each step is based on the coefficients estimation, using a variant of BCR, followed by the update of one chosen basis, using Singular Value Decomposition. We assess experimentally how well the learning algorithm recovers dictionaries that may or may not have the required structure, and to what extent the noise level is a disturbing factor.

1. INTRODUCTION

Sparse coding [1, 2] is a useful tool to analyze and try to explain the structure of series of observed data, such as successive time frames of an audio signal [3] or natural images [4]. Formally, assume that we observe T vectors $\mathbf{x}(t) = (x_n(t))_{n=1}^N, 1 \leq t \leq T$ which are supposed to have been generated following the model:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \boldsymbol{\epsilon}(t) \tag{1}$$

$\mathbf{A}$ being an overcomplete dictionary (an $N \times K$ matrix, with $K \geq N$), $\mathbf{s}(t) \in \mathbb{R}^K$ some "sparse" coefficients and $\boldsymbol{\epsilon}(t) \in \mathbb{R}^N$ a Gaussian noise. Sparse coding can be viewed as a way of estimating $\mathbf{A}$ from the only observation of $\mathbf{X} = \mathbf{AS} + \mathbf{E}$ where $\mathbf{X}$ is the $N \times T$ matrix containing T signal frames (similar notations being used for $\mathbf{S}$ and $\mathbf{E}$).

Jointly optimizing the coefficients and the dictionary, under constraints added to enforce the well-posedness of the problem, is a hard task, so we use an alternating optimization strategy:

1. **Coefficient update** given a dictionary $\mathbf{A}$:

$$\arg\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{AS}\|_2^2 + \lambda\|\mathbf{S}\|_1 \tag{2}$$

2. **Dictionary update** given coefficients $\mathbf{S}$:

$$\arg\min_{\mathbf{A}} \|\mathbf{X} - \mathbf{AS}\|_2^2 \tag{3}$$

under some constraint on $\mathbf{A}$.

The coefficient update step (2) can be justified in a probabilistic framework using a Laplacian prior on the coefficients $s_k(t)$ [5]. Moreover, it is a simpler parent problem of the NP-hard combinatorial problem, where $\|\mathbf{S}\|_1$ is replaced with $\|\mathbf{S}\|_0^0$, the number of non-zero components in $\mathbf{S}$. Computing the solution to Eq. (2) by Quadratic Programming is rather computationally intensive in the general case where $\mathbf{A}$ has no special structure. However, when $\mathbf{A}$ is a union of orthonormal bases (ONB), Block Coordinate Relaxation (BCR) methods are efficient [6]. Another motivation to constrain the dictionary to be a union of ONB is that it seems that audio signals [7] and images [8] can indeed be modeled as the superimposition of several layers, each of which having a sparse representation in its own adapted ONB. Note that when $\mathbf{A}$ is constrained to have this precise structure, the dictionary update step (3) is also made relatively easy. This step, in a probabilistic framework, can be interpreted as a likelihood maximization and solved with an Expectation-Maximization (EM) algorithm [9].

In Section 2, we describe BCR and a variant which we used to solve (2). In Section 3, we introduce our algorithm to learn a union of bases by iteratively optimizing (3) with respect to each basis. In Section 4 we describe and analyze the experiments made with the learning algorithm on data generated following the model (1). We study the influence of the number T of frames of the learning dataset, of the *a priori* knowledge on the noise level, and of the possible modeling error corresponding to the fact that the true $\mathbf{A}$ might not be a union of bases or the number of bases could be wrong.

⋆The initial part of this work was done in collaboration with Laurent Benaroya while he was finishing his PhD with IRISA.

2. COMPUTATION OF SPARSE COEFFICIENTS

Finding sparse coefficients for the observed data $\mathbf{X}$ is the result of a compromise between

- the minimization of the **reconstruction error**:

$$\|\mathbf{X} - \mathbf{AS}\|_2^2 := \sum_{n=1}^{N} \sum_{t=1}^{T} |x_n(t) - (\mathbf{As}(t))_n|^2.$$

- the minimization of a **diversity** measure, the most common ones being:

$$\|\mathbf{S}\|_p^p := \sum_{k=1}^{K} \sum_{t=1}^{T} |s_k(t)|^p$$

for $0 \leq p \leq 1$. The strict diversity, defined by the number of non-zeros coefficients, is given by $\|\mathbf{S}\|_0^0$.

This problem is generally difficult. It is indeed NP-hard with the $\|\mathbf{S}\|_0^0$ diversity measure when $\mathbf{A}$ is an arbitrary redundant dictionary. Many sub-optimal algorithms have been proposed such as Matching Pursuit (MP) [10], Basis Pursuit (BP) [5] and FOCUSS [11]. The latter solves the minimization problem

$$\min_{\mathbf{S}} \|\mathbf{X} - \mathbf{AS}\|_2^2 + \lambda \|\mathbf{S}\|_p^p \tag{4}$$

and Basis Pursuit solves it for $p = 1$ (see Eq. (2)). These algorithms are generally rather computationally intensive. However Basis Pursuit can be implemented more efficiently with a Block Coordinate Relaxation (BCR) method [6] when $\mathbf{A}$ is a union of ONB. Moreover, it has been shown that, under some conditions, the solution of (4) for any $0 \leq p \leq 1$ is close to the solution given by Basis Pursuit [12].

In this section we recall how Basis Pursuit is implemented with soft-thresholding when $\mathbf{A}$ is a single ONB, then we remind the reader about BCR and eventually we describe a variant of BCR that we introduced to deal with low noise levels (small threshold parameter λ).

2.1. Case of an orthonormal basis

When $\mathbf{A}$ is an orthonormal basis, the solution of (2) is given by soft thresholding:

$$\forall k, t \; \hat{s}_k(t) = \begin{cases} \mathbf{a}_k^T \mathbf{x}(t) - \lambda/2 & \text{if } \mathbf{a}_k^T \mathbf{x}(t) > \lambda/2 \\ 0 & \text{if } |\mathbf{a}_k^T \mathbf{x}(t)| \leq \lambda/2 \\ \mathbf{a}_k^T \mathbf{x}(t) + \lambda/2 & \text{if } \mathbf{a}_k^T \mathbf{x}(t) < -\lambda/2 \end{cases} \tag{5}$$

where $\mathbf{a}_k$ is the k^{th} column of $\mathbf{A}$ (also called *atom* of the dictionary).

2.2. Case of a union of orthonormal bases

When $\mathbf{A} = [\mathbf{A}_1, \ldots, \mathbf{A}_L]$ is a union of L orthonormal bases, the coefficients $\mathbf{S}$ are decomposed in L subsets of coefficients $\mathbf{S}_l$ corresponding to the L bases, as $\mathbf{S} = [\mathbf{S}_1^T, \ldots, \mathbf{S}_L^T]^T$. The BCR algorithm, described in Table 1 deals with the difficulty to directly solve (2) for a redundant dictionary $\mathbf{A}$ by successively solving it for its different bases $\mathbf{A}_l$. Then the sub-coefficients of an initial estimate $\mathbf{S}_{init}$ are iteratively updated until convergence is reached. The BCR algorithm has been proven to converge in a wake sense to a solution of (2), when, in Step 1, the selection of $\mathbf{S}_l$ follows a systematic cycle, or results from an optimal descent rule [6]. Unfortunately, if λ is very small (which corresponds to looking for a small reconstruction error, namely the low noise assumption), BCR might converge very slowly since almost no thresholding is performed in Step 3. In order to compute sparse coefficients in this low noise case, we propose to start BCR with a large initial threshold λ_0 and decrease it regularly, leading to the algorithm explained in Table 2. During the very first steps, since the threshold is high, sparsity is enforced; when the threshold becomes smaller, the error vanishes.

1. Select a subset $\mathbf{S}_l$ of the current $\mathbf{S}$ to update;
2. Compute $\mathbf{X}_l = \mathbf{X} - \sum_{i \neq l} \mathbf{A}_i \mathbf{S}_i$;
3. Update $\mathbf{S}_l$ by replacing it by

$$\arg\min_{\mathbf{S}_l} \|\mathbf{X}_l - \mathbf{A}_l \mathbf{S}_l\|_2^2 + \lambda \|\mathbf{S}_l\|_1,$$

which is computed by soft thresholding (Eq. (5));

4. If the stopping criterion is not reached, go to step 1.

Table 1. Block Coordinate Relaxation algorithm

for $it = 0$ to N_{it}

use BCR with threshold $\lambda_0(1 - \frac{it}{N_{it}})$ to update $\mathbf{S}$

end

Table 2. Modified BCR algorithm for the low noise case

2.3. Experiments

Even though we have no proof of convergence of this modified BCR algorithm, we observed experimentally that if the two parameters N_{it} and λ_0 are appropriately chosen, it reaches a solution close to those provided at a higher computational cost by MP and FOCUSS.

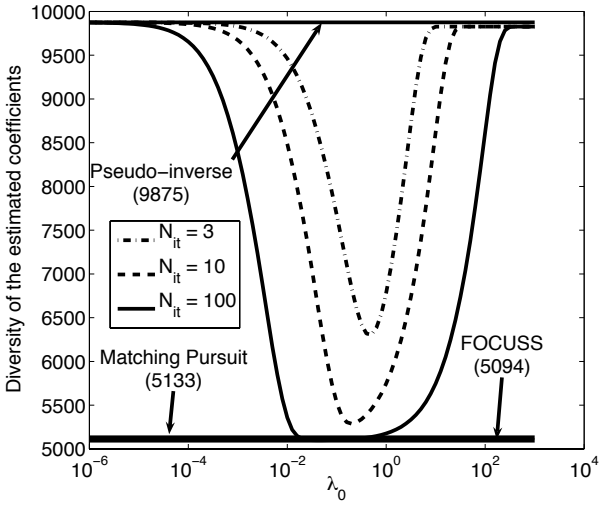

Fig. 1. Diversity of the coefficients computed by the BCR variant, depending on the initial threshold, and the number of iterations

We ran the following experiment. Data are sparsely generated from a dictionary that is the union of two orthonormal bases. Using this dictionary, the BCR variant was run on the data, for different values of the initial threshold (between 10^{-6} and 10^{3}), and for different number of iterations (3, 10 and 100). The diversity of the obtained coefficients is displayed in figure 1. We can see that, for any N_{it}, the initial threshold λ_0 greatly impacts the diversity of the estimated coefficients. An optimal value of λ_0 may be choosen *a posteriori* to minimize (2). Note that the higher the number of iterations, the sparser the coefficients obtained using the optimal threshold.

For $N_{it} = 100$ and λ_0 in a fairly large range, the diversity of the coefficients obtained by the BCR variant is the same than for MP and FOCUSS. However, the BCR algorithm variant is computationally less costly, taking about 3.5 seconds instead of about 9 seconds for FOCUSS, and 10 seconds for MP.

3. DICTIONARY LEARNING WITH SVD

The algorithm used to learn a union of L orthonormal bases (ONB) is described in Table 3. To understand the rationale behind the use of the SVD in Step 3, we will first analyze the optimization problem (3) when $\mathbf{A}$ is constrained to be a single ONB. Then, we will briefly explain how the algorithm for L bases is derived from the single basis one, and we will discuss in more details how the coefficient update (Step 2) is performed, depending whether we know which value of λ to use in (Step 2) or we want to adapt it to the data. As for the the stopping criterion (Step 4), we simply set *a priori* the number of learning steps. Studying how much the dictionary varies between two steps may help design a better criterion in the future.

1. Choose an initial dictionary $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_L]$;
2. Update the coefficients $\mathbf{S} = [\mathbf{S}_1 \dots \mathbf{S}_L]$ using the current $\mathbf{A}$ (see text);
3. Choose which basis $\mathbf{A}_l$ to update and:
   (a) Compute $\mathbf{X}_l = \mathbf{X} - \sum_{i \neq l} \mathbf{A}_i \mathbf{S}_i$
   (b) Compute a singular value decomposition:
   $$\mathbf{X}_l \mathbf{S}_l^T = \mathbf{U}\mathbf{D}\mathbf{V}^T$$
   (c) Update
   $$\mathbf{A}_l = \mathbf{U}\mathbf{V}^T$$
4. If the stopping criterion is not reached, go to step 2 (see text).

Table 3. Learning algorithm for L orthonormal bases

3.1. Learning a single orthonormal basis

The optimization problem (3) with the constraint that $\mathbf{A}$ is an ONB can be written as the minimization of a Lagrangian,

$$\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) = \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_2^2 + \mathrm{Tr}\left[\boldsymbol{\mu}(\mathbf{A}^T\mathbf{A} - \mathbf{Id})\right] \qquad (6)$$

where $\boldsymbol{\mu}$ is an $N \times N$ matrix of Lagrange multipliers chosen so that the minimizing matrix $\mathbf{A}_{opt}$ is an ONB.

The optimal dictionary $\mathbf{A}_{opt}$, minimizing the Lagrangian, is obtained as follows (the proof is in annex A):

Proposition 3.1 *Let $\mathbf{U}\mathbf{D}\mathbf{V}^T$ be (one of) the Singular Value Decomposition(s) (SVD) of $\mathbf{X}\mathbf{S}^T$, that is to say $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{D}$ is a diagonal matrix. An optimal ONB is given by $\mathbf{A}_{opt} = \mathbf{U}\mathbf{V}^T$. If $\mathbf{X}\mathbf{S}^T$ is invertible, it is the unique optimal solution.*

Note that, when $\mathbf{X}\mathbf{S}^T$ is invertible, the product $\mathbf{U}\mathbf{V}^T$ does not depend on the particular choice of a SVD $\mathbf{U}\mathbf{D}\mathbf{V}^T$.

3.2. Learning a union of L orthonormal bases

Ideally, when $\mathbf{A}$ is constrained to be the union of L orthonormal bases, one would like to perform the dictionary update step (3) by minimizing the Lagrangian:

$$\|\mathbf{X} - \sum_{l=1}^{L} \mathbf{A}_l \mathbf{S}_l\|_2^2 + \sum_{l=1}^{L} \mathrm{Tr}\left[\boldsymbol{\mu}_l(\mathbf{A}_l^T\mathbf{A}_l - \mathbf{Id})\right].$$

However, this optimization problem does not have an explicit solution as in the case of a single ONB. The principle

behind the algorithm described in Table 3 is that, at each iteration, only one of the bases $\mathbf{A}_l$ is optimized.

If we know which parameter λ to use in Eq. (2) – for example, if we know the prior distributions of ϵ and $\mathbf{s}$ in the probabilistic model (1) – then we perform Step 2 with the regular BCR algorithm. In many practical cases however, it is difficult to have an idea of a relevant value for λ, or in the low noise limit, λ is too small. Below we discuss how Step 2 in Table 3 is performed depending on the noise model.

3.2.1. Unknown gaussian noise

If Gaussian noise $\boldsymbol{\epsilon}(t)$ is assumed, with unknown variance, we use the algorithm proposed by Azzalini *et al*, [13]: starting from an exact representation $\mathbf{X} = \mathbf{AC}$, with $\mathbf{A}$ the current dictionary, and the estimate $\mathbf{S} = 0$ we iterate the following steps

1. compute the variance σ^2 of the residual $\mathbf{R} = \mathbf{C} - \mathbf{S}$;
2. update $\mathbf{S}$ by letting it contain all the entries of $\mathbf{R}$ that are above the threshold $\lambda = \sqrt{2\log(N)\sigma^2}$.
3. if the last update did not modify $\mathbf{S}$, then stop, else go to 1.

To compute the exact decomposition $\mathbf{C}$, rather than using $\mathbf{C} = \mathbf{A}^+\mathbf{X}$, one can use indifferently FOCUSS, MP, or the variant of BCR, because they all encourage sparsity of the coefficients.

3.2.2. Low noise limit

In the low noise limit, each of the above strategies for Step 2 will essentially fail since the dictionary update step will (almost) not change the dictionary.

To see it, let us simply analyse the first iteration of the algorithm described in Table 3.

Given the initial dictionary $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_L]$ and the data $\mathbf{X}$, the coefficients $\mathbf{S} = [\mathbf{S}_1 \dots \mathbf{S}_L] = \mathbf{S}(\mathbf{A}, \mathbf{X})$ computed at Step 2 are uniquely defined. Because of the low noise assumption, they give a perfet reconstruction $\mathbf{X} = \mathbf{AS}$.

Then, if $\mathbf{X}_l\mathbf{S}_l^T$ is invertible, the proposition 3.1 shows that the optimal basis $\hat{\mathbf{A}}_l$ which minimizes the Lagrangian (6) is unique. And, for the current basis $\mathbf{A}_l$, the first term in the Lagrangian – the error term – is null, and the second term is null too, because of the orthonormality of the basis. The lower bound of the Lagrangian for an orthonormal basis is then reached by the current basis. We conclude that $\hat{\mathbf{A}}_l = \mathbf{A}_l$ and there can be no change.

If $\mathbf{XS}^T$ is not invertible, the optimal basis may be different, but this case does not appear experimentally.

Thus, in the low noise limit, we propose to perform Step 2 with a heuristic strategy that allows the dictionary to be modified at each step. To do that, we add some reconstruction error at each step by thresholding the coefficients. The added reconstruction error is decreased each time Step 2 is performed so that at the end of the process we get exact reconstruction.

1. Choose an initial dictionary $\mathbf{A} = [\mathbf{A}_1 \dots \mathbf{A}_L]$ and set N_{kept}, the number of kept coefficients by frame, to zero;
2. Update the coefficients $\mathbf{S} = [\mathbf{S}_1 \dots \mathbf{S}_L]$ using the current $\mathbf{A}$, by the BCR variant;
3. If the algorithm stagnates, increment N_{kept};
4. Set all but the $N_{kept}T$ greatest coefficients of $\mathbf{S}$ to zero, giving $\mathbf{S}^{thres}$;
5. Choose which basis $\mathbf{A}_l$ to update and:
   (a) Compute $\mathbf{X}_l = \mathbf{X} - \sum_{i \neq l} \mathbf{A}_i\mathbf{S}_i^{thres}$
   (b) Compute a singular value decomposition:
   $$\mathbf{X}_l{\mathbf{S}_l^{thres}}^T = \mathbf{UDV}^T$$
   (c) Update
   $$\mathbf{A}_l = \mathbf{UV}^T$$
6. If the stopping criterion is not reached, go to step 2.

Table 4. Learning algorithm for L orthonormal bases in the low noise limit

Since the learning (experimentally) leads to decrease the diversity of the coefficients $\mathbf{S}$ along the iterations, the algorithm is said to stagnate when this diversity stops decreasing. By example, it can be decided the algorithm stagnates when the relative difference between the diversities of the five last coefficients matrices is least than 10^{-3}.

4. EXPERIMENTS

The following experiments have been designed with synthetic data generated with the model (1) using a known reference dictionary $\mathbf{A}_{ref}$. The goal is to study the influence of various parameters on the performance, and to see how modeling error could also impact the results. We use two relevant and complementary performance measures: the false alarm rate and the missed detection rate, corresponding respectively to the relative number of estimated atoms $\mathbf{a}_{est}$ (i.e. columns of $\mathbf{A}_{est}$, the dictionary estimated using the learning algorithm) that "do not match" any reference atom $\mathbf{a}_{ref}$, and to the relative number of reference atoms that "are not matched" by any estimated atom. Since all atoms have unit norm, $\mathbf{a}_{est}$ and $\mathbf{a}_{ref}$ are considered to match if their

inner product $|\mathbf{a}_{est}^T \mathbf{a}_{ref}|$ is close enough to one. So we use a parameter ξ to decide that they match if and only if $|\mathbf{a}_{est}^T \mathbf{a}_{ref}| \geq \xi$. Different values of ξ yield different but related performance measures.

In order to evaluate the performance on a wide range of conditions, each experiment is run with N_r different dictionaries, and the performance measures are averaged over these runs.

4.1. Influence of the number T of signal frames

First, we study the impact of the number T of frames used to learn the dictionary, for different dimensions N. Data are generated with $\mathbf{A}_{ref}$ a union of two random ONB, using the noiseless model (1) with $\mathbf{s}(t)$ (of dimension $2N$) containing between 1 and r, randomly located, non-zero coefficients, that follow a standard Gaussian law. r is depending on the dimension N and is chosen as follows: $r = \lfloor (1+\sqrt{N})/2 \rfloor$. Indeed, for this number of non-zero coefficients, the generating $\mathbf{s}(t)$ are the only coefficients, among those giving exact decomposition, that solve (2) [12].

The experiments are performed for the following dimensions : $N = 4, 8, 16, 32, 64$, and, for each of them, for $T = N/4, N, 2N, 5N, 10N, 20N, 50N, 100N$.

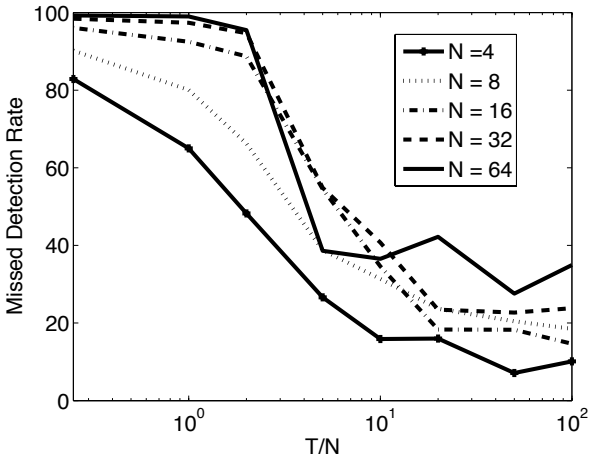

Fig. 2. Average Missed Detection Rate (on $N_r = 100$ runs) depending on the number of frames T for different dimensions N

Figure 2 displays the average Missed Detection Rate of the learned dictionary (average on $N_r = 100$ runs) depending on the number of frames T. The False Alarm Rate is not displayed since it is nearly the same.

The MD Rate is decreasing as the relative number of learning frames is growing. For $T \leq 2N$, the dictionary estimation yields poor performance, the false alarm rate and the missed detection rate are both greater than 50%, for $\xi = 0.99$, and even than 90%, for $N \geq 16$. On the contrary, for $T = 100N$, about seven out of ten atoms are correctly estimated. Noting that the computing time greatly increases with T, we set $T = 50N$ in the rest of the experiments.

4.2. Influence of the noise level

To understand the effect of the noise level on the performance, we repeat the above experiments with the same data to which we add noise at various signal to noise ratios (SNR): $+\infty$ dB, 10 dB, 0 dB. At each SNR level we run the three configurations of the learning algorithm designed respectively for known λ, small λ and unknown λ.

Learning algorithm	$+\infty$ dB	$+10$ dB	$+0$ dB
λ_{known}	7%	28%	59%
λ_{small}	6%	30%	63%
$\lambda_{unknown}$	42%	58%	86%

Table 5. Missed Detection Rate depending on the noise level on data, and on the learning method, with $\xi = 0.99$ (average over $N_r = 60$ experiments)

Table 4.2 shows that the algorithms with known λ, and small λ (without prior knowledge on λ), give almost similar results, the first one performing better when there is some noise, while the second one giving better estimation in the low-noise case. Unfortunately, the algorithm designed for unknown λ always misses more atoms. Thus, in the following, this algorithm by Azzalini will no more be used.

The three configurations of the algorithm are greatly dependent on the noise level. Indeed, even for a $+10$dB signal to noise ratio, among the $N_r = 60$ runs, the Missed Detection Rate of the most succesful experiment is no smaller than 14%. This means that for this SNR level, the algorithms never finds more than 86% of the atoms of the dictionary.

On the contrary, when there is no noise, the first two algorithms retrieve all the atoms in most of the runs. A part of the experiments entirely fail, leading to the means displayed in Table 4.2 (7% and 6%).

4.3. Influence of the dimension N of signal frames

In order to understand the influence of the dimension N of signal frames on the behaviour of the algorithm, we run the learning for noiseless data, as in subsection 4.1, for different values of N. The dictionary is made of two random ONB.

The first experiment illustrates how many iterations are required for the algorithm to converge, depending on N. For each N, $N_r = 100$ experiments are launched, and results are then averaged. The algorithm is decided to converge when the original dictionary has been retrieved with a sufficient precision, namely when the Missed Detection Rate,

for a threshold $\xi = 0.99$, reaches 0%. The size of frames N is chosen between 4 and 128. Indeed, for higher values of N, the computation is too long, and learning is unfeasible.

The figure 3 shows the number of iterations needed to converge, depending on N, when the algorithm converges. The N_r points, for each N, represent the number of iterations needed to converge in each of the N_r experiments. The circles represent the mean for a given N, and the bars, the associated standard deviation. The broken line is the linear regression of the means.

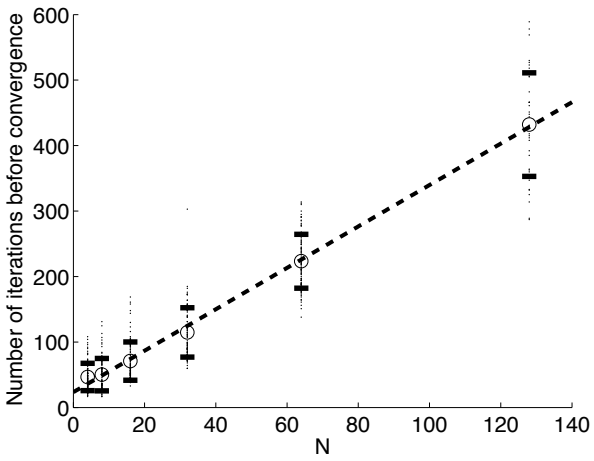

Fig. 3. Average number of iterations before convergence, depending on N, in the cases when the algorithm effectively converge

We see that, in the case when the learning is succesful, the average number of iterations to reach convergence increases linearly with the size N. Using the mean is accurate since the variance of the data points around the mean is small enough to clearly see the linear dependence between N and the number of iterations before convergence.

The figure 4 shows the percentage of converging experiments.

We see that, for very small values of N, very few experiments converge, and the algorithm seems also to be less efficient when N become high. Work should be done to decide, for a given experiment, if the algorithm has converged to a correct solution. An idea would be to learn three dictionaries and to make a vote, hoping that two dictionaries out of the three are the good ones.

A second experiment is performed, showing the influence of the size of the signal frames on the computational cost. The number of samples N_{total} in the whole signal is set constant (i.e. for different N, the signals are different frame hashings of the same original signal). The number T of frames is then $T = \frac{N_{total}}{N}$. Note we could expect bad learning results when T become too small, according to

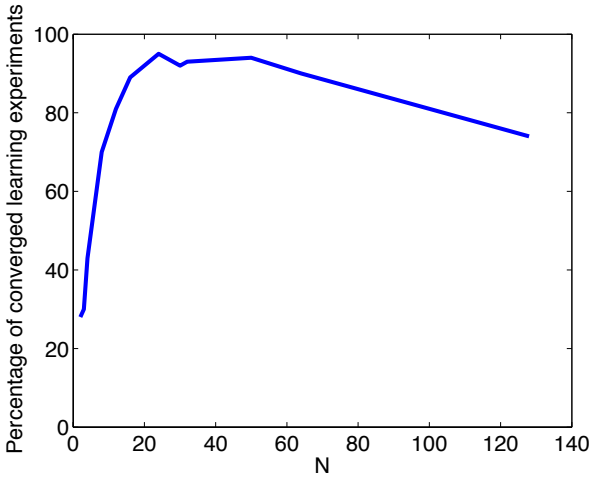

Fig. 4. Percentage of experiments where the algorithm have converged (the estimated dictionary is the original one)

subsection 4. In fact, in this experiment, we only look at the computational time, and learning is not performed until the end (and the result of the learning is not important here).

The average computational time of each step of the algorithm in Table 3 is measured. As a result, we observe that for high values of N (from $N = 128$ to 2048), the computational time of the SVD Step 3b, the matrices multiplication Steps 3a and 3c and the coefficients update Step 2 grow as a power of N (between $O(N^2)$ and $O(N^3)$). Thus, on a Pentium 4, for $N = 512$, a single iteration takes $10s$, and for $N = 2048$, it takes $324s$.

The computational time for performing learning then increases for two reasons as N grows. Firstly, it increases due to the higher computational cost of each step, and secondly, more iterations are needed before convergence. The algorithm is then untractable for values of N greater than 64.

But, in audio signal, the frame size is commonly about $20ms$. At a sampling frequency of $44100Hz$, a frame of $23.2ms$ corresponds to $N = 1024$, that is untractable with this algorithm. A challenge would be to adapt the method to high dimension frames.

4.4. Influence of the model mismatch

We designed experiments to analyse the behavior of the learning algorithm when there is a mismatch between the number of ONB in the reference dictionary and in the estimated one.

We run the same algorithm to estimate a pair of ONB on three datasets generated as above with the noiseless model (1) with

- $\mathbf{A}_{ref,1}$ a single random ONB
- $\mathbf{A}_{ref,2}$ a union of two ONB

- $\mathbf{A}_{ref,3}$ a union of three random ONB.

Note that with $\mathbf{A}_{ref,3}$, if the three bases are sufficiently different one from another, one cannot expect to get less than 33% missed detection, because only $2N$ atoms are estimated while there are $3N$ reference atoms. With $\mathbf{A}_{ref,1}$, as soon as there is no more than 50% false alarm we are sure to have recovered the atoms of the reference dictionary, but they may be split in the two learned bases.

Reference dictionary	$\mathbf{A}_{ref,1}$	$\mathbf{A}_{ref,2}$	$\mathbf{A}_{ref,3}$
Average missed detection rate	0.5%	7%	99.5%
Average false alarm rate	44%	7%	99%

Table 6. missed detection rate and false alarm rate ($\xi = 0.99$) depending on the number of bases in $\mathbf{A}_{ref}$, when the estimated dictionary owes two bases (average over $N_r = 200$ runs)

The results of two hundred runs are summarized in Table 4.4. More precisely:

- $\mathbf{A}_{ref,1}$: almost all reference atoms are retrieved. In 55 cases out of 100, one of the estimated basis is exactly $\mathbf{A}_{est,1}$. In the 45 other cases out of 100, the retrieved atoms are shared between the two bases, 82% in the first basis, and 17% in the second, while 1% are not detected.

- $\mathbf{A}_{ref,2}$: the average performance values hide two distinct behaviours. In 92% of the experiments, the dictionary is perfectly estimated, while in 8% of the cases, learning totally failed, without any well estimated atom at all.

- $\mathbf{A}_{ref,3}$: never more than 15% of the atoms were retrieved, the average being only 0.5%.

The algorithm seems to be efficient only when estimating at least as many bases as there are in the reference dictionary. A good strategy could therefore be to learn a lot of bases, and to estimate *a posteriori* the number of interesting ones.

5. CONCLUSION

We have presented a new method for learning, from a set of observed data vectors, a dictionary structured as a union of orthonormal bases, with the objective that the decomposition of the data on this dictionary would be sparse. We have demonstrated on synthetically generated data that this method is able to recover a relevant underlying dictionary provided that it knows *a priori* the structure (i.e. the number of ONB in the dictionary). The approach seems to behave reasonably well even when the number of bases is overestimated. We are now considering several remaining practical problems, namely estimating the number of bases, studying how the algorithm scales when the dimension N of the data becomes large and extending experiments to real audio signals or images. Last but not least, we are looking for conditions where we can prove that the algorithm converges to the underlying dictionary.

APPENDIX

A. PROOF OF PROPOSITION 3.1

Let detail the proof giving the optimal solution $\mathbf{A}_{opt}$ to the minimization of the Lagrangian (6). The Lagragian is:

$$\mathcal{L}(\mathbf{A}, \boldsymbol{\mu}) = \|\mathbf{X} - \mathbf{AS}\|_2^2 + \mathrm{Tr}\left[\boldsymbol{\mu}(\mathbf{A}^T\mathbf{A} - \mathbf{Id})\right]$$

The optimal dictionary $\mathbf{A}_{opt}$ necessarily verify that the gradients $\nabla_{\boldsymbol{\mu}}\mathcal{L}$ and $\nabla_{\mathbf{A}}\mathcal{L}$ are equal to zero:

$$\mathbf{A}^T\mathbf{A} - \mathbf{Id} = 0 \quad (7)$$

$$-2(\mathbf{X} - \mathbf{AS})\mathbf{S}^T + \mathbf{A}(\boldsymbol{\mu} + \boldsymbol{\mu}^T) = 0 \quad (8)$$

or

$$\begin{cases} \mathbf{A}^T\mathbf{A} = \mathbf{Id} \\ \mathbf{XS}^T = \mathbf{A}\left[\mathbf{SS}^T + \frac{1}{2}(\boldsymbol{\mu} + \boldsymbol{\mu}^T)\right] \end{cases} \quad (9)$$

Let $\mathbf{Z} := \mathbf{XS}^T$ and $\mathbf{Y} := \mathbf{SS}^T + (\boldsymbol{\mu} + \boldsymbol{\mu}^T)/2$. While $\mathbf{Z}$ can be explicitly computed, $\mathbf{Y}$ is unknown since it depends on the unknown multipliers $\boldsymbol{\mu}$.

The system (9) implies:

$$\mathbf{ZZ}^T = \mathbf{AYY}^T\mathbf{A}^T = \mathbf{AY}^2\mathbf{A}^T$$

because $\mathbf{Y}$ is symmetric, and

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{Y}^T\mathbf{A}^T\mathbf{AY} = \mathbf{Y}^2$$

because of the orthogonality of $\mathbf{A}$. Then, combining the two preceding results, $\mathbf{A}_{opt}$ must satisfy:

$$\mathbf{AZ}^T\mathbf{ZA}^T = \mathbf{ZZ}^T \quad (10)$$

Let $\mathbf{Z} = \mathbf{UDV}^T$ be one of the possible Singular Value Decompositions (SVD) of $\mathbf{Z}$, that is to say $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices and $\mathbf{D}$ is a diagonal matrix. We assume that the diagonal elements of $\mathbf{D}$ (the singular values) are ordered decreasingly, i.e. noting

- L the number of different singular values in $\mathbf{D}$,

- N_k the multiplicity of the lth singular value λ_k,

- $\mathbf{I}_k$ the identity matrix of size N_l

we have $\lambda_1 > \lambda_2 > \ldots > \lambda_L \geq 0$, and

$$\mathbf{D} = \begin{bmatrix} \lambda_1 \mathbf{I}_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_L \mathbf{I}_L \end{bmatrix}$$

Note that between two different SVD, only the orthogonal matrices differs, while the diagonal matrix $\mathbf{D}$ is the same.

Then equation (10) gives:

$$\mathbf{A}\mathbf{V}\mathbf{D}^2\mathbf{V}^T\mathbf{A}^T = \mathbf{U}\mathbf{D}^2\mathbf{U}^T$$

and noting $\mathbf{R} = \mathbf{U}^T\mathbf{A}\mathbf{V}$ (equivalent to $\mathbf{A} = \mathbf{U}\mathbf{R}\mathbf{V}^T$), $\mathbf{R}$ is orthonormal and :

$$\mathbf{R}\mathbf{D}^2 = \mathbf{D}^2\mathbf{R} \tag{11}$$

Let note:

$$\mathbf{R} = [\mathbf{R}_{ij}]_{i,j}$$

where $\mathbf{R}_{ij}$ is a N_i by N_j submatrix.

As $\mathbf{R}$ commutes with $\mathbf{D}^2$ in (11), we have:

$$\mathbf{R}\mathbf{D}^2 = \left[\lambda_j^2\mathbf{R}_{ij}\right]_{ij} = \mathbf{D}^2\mathbf{R} = \left[\lambda_i^2\mathbf{R}_{ij}\right]_{ij}$$

and then, for all $i \neq j$,

$$\lambda_j^2\mathbf{R}_{ij} = \lambda_i^2\mathbf{R}_{ij}$$

Since $\lambda_i^2 \neq \lambda_j^2$, $\mathbf{R}_{ij} = \mathbf{0}$ for all $i \neq j$, and $\mathbf{R}$ is block-diagonal:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \mathbf{R}_{LL} \end{bmatrix}$$

where for each l, the block $\mathbf{R}_{ll}$ is an orthogonal matrix.

The functional (6) must then have the following form :

$$\mathcal{L}(\mathbf{A}) = \|\mathbf{X} - \mathbf{A}\mathbf{S}\|_2^2 = \text{Tr}(\mathbf{X}^T\mathbf{X}) + \text{Tr}(\mathbf{S}^T\mathbf{S}) - 2\text{Tr}(\mathbf{X}\mathbf{S}^T\mathbf{A}^T)$$

The first two terms are independent of $\mathbf{A}$, then one must maximize $\text{Tr}(\mathbf{X}\mathbf{S}^T\mathbf{A}^T)$. Rewriting it gives

$$\begin{aligned} \text{Tr}(\mathbf{X}\mathbf{S}^T\mathbf{A}^T) &= \text{Tr}(\mathbf{Z}\mathbf{A}^T) \\ &= \text{Tr}(\mathbf{U}\mathbf{D}\mathbf{R}^T\mathbf{U}^T) \\ &= \text{Tr}(\mathbf{D}\mathbf{R}^T) \\ &= \text{Tr}(\mathbf{R}\mathbf{D}) \\ &= \sum_{l=1}^{L} \lambda_l \text{Tr}(\mathbf{R}_{ll}) \end{aligned} \tag{12}$$

One must then maximize each of the terms of the sum. For each l such that $\lambda_l > 0$, one must maximize $\text{Tr}(\mathbf{R}_{ll})$.

As all the diagonal elements of an orthonormal matrix are lower than 1, the maximal trace is N_l. It is only reached by the identity matrix, which is the only orthogonal matrix all diagonal elements of which are ones.. The trace of $\mathbf{R}_{ll}$ is then maximal if and only if $\mathbf{R}_{ll}$ is the identity matrix. If the last singular value λ_L is null, any orthogonal matrix $\mathbf{R}_{LL}$ gives $\lambda_L\text{Tr}(\mathbf{R}_{LL}) = 0$, and then is suitable.

The best matrix $\mathbf{R}$ is then the identity matrix, and it's the only one, in the case where all the singular values are strictly positive. When the last singular values are null, the identity matrix is again optimal, but no longer unique.

The optimal dictionary $\mathbf{A}$, for a given SVD is then the following

$$\mathbf{A}_{opt} = \mathbf{U}\mathbf{V}^T \tag{13}$$

Remarking that the product $\mathbf{U}\mathbf{V}^T$ is independent from the chosen SVD, this matrix is optimal, for every chosen decomposition.

Let us prove that $\mathbf{U}\mathbf{V}^T$ is independent of the SVD.

JE VAIS REGARDER A LA BIBLIOTHEQUE POUR UN BOUQUIN ETABLISSANT CE RESULTAT

Let $\mathbf{U}\mathbf{D}\mathbf{V}^T$ and $\mathbf{L}\mathbf{D}\mathbf{M}^T$ be two different singular value decompositions of the matrix $\mathbf{Z}$. Then,

$$\mathbf{Z}^T\mathbf{Z} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T = \mathbf{M}\mathbf{D}^2\mathbf{M}^T$$

Noting $\mathbf{T} = (\mathbf{V}^T\mathbf{M})$, this is equivalent to:

$$\mathbf{T}\mathbf{D}^2 = \mathbf{D}^2\mathbf{T}$$

As before, the matrix $\mathbf{T}$ is then block-diagonal. Moreover, as $\mathbf{T}$ is orthogonal, every diagonal block is orthogonal.

The first result is that two different SVD are related by:

$$\mathbf{M} = \mathbf{V}\mathbf{T}$$

and

$$\mathbf{L} = \mathbf{U}\mathbf{T}$$

with $\mathbf{T}$ a block-diagonal matrix of orthogonal matrices, each of size of the multiplicity of the corresponding singular value.

The second result is, for the same two SVD, that the products $\mathbf{U}\mathbf{V}^T$ and $\mathbf{L}\mathbf{M}^T$ are equal:

$$\mathbf{L}\mathbf{M}^T = \mathbf{U}\mathbf{T}\mathbf{T}^T\mathbf{V}^T = \mathbf{U}\mathbf{V}^T$$

since $\mathbf{T}$ is an orthogonal matrix.

B. REFERENCES

[1] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural computation*, vol. 12, no. 2, pp. 337–365, 2000.

[2] M.S. Lewicki and B. Olshausen, "A probabilistic framework for the adaptation and comparison of image codes," *Journal of the Optical Society of America*, 1999.

[3] S.A. Abdallah and M.D. Plumbley, "If edges are the independent components of natural images, what are the independent components of natural sounds?," in *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*, december 2001, pp. 534–539.

[4] A.J. Bell and T.J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vision research*, vol. 37, no. 23, pp. 3327–3338, 1997.

[5] S.S. Chen, D.L. Donoho, and M.A. Saunder, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[6] S. Sardy, A.G. Bruce, and P. Tseng, "Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries," *Journal of computational and graphical statistics*, vol. 9, pp. 361–379, 2000.

[7] S. Molla and B. Torresani, "Determining local transientness in audio signals," *IEEE signal processing letters*, vol. 11, no. 7, pp. 625–628, july 2004.

[8] J.L. Starck, M. Elad, and D.L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE transactions on image processing*, february 2004.

[9] T.K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, pp. 47–60, november 1996.

[10] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[11] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, "Dictionary learning algorithms for sparse representation," *Neural Computation*, vol. 15, pp. 349–396, 2003.

[12] R. Gribonval and M. Nielsen, "On the strong uniqueness of highly sparse representations from redundant dictionaries," in *Proceedings of the fifth International Conference on Independent Component Analysis and Blind Signal Separation*, september 2004, pp. 201–208.

[13] A. Azzalini, M. Farge, and K. Schneider, "A recursive algorithm for nonlinear wavelet thresholding : Application to signal and image processing," Tech. Rep. 41, Institut Pierre Simon Laplace, 2004.